CSS셀렉터, 기초 이론 정리, 포트폴리오 만드는 날까지! |

프론트엔드 개발자 입문편: HTML, CSS, Javascript

**1. 의미, 정의**

**CSS(Cascading Style Sheet)**

예)기본적으로 작성된 Style Sheet이 아닌 스타일링을 작성한 문서가 있다면 그것을 사용하는 것을 뜻한다.

웹 사이트를 스타일링할 때 대표적으로 세 가지로 나뉜다.

Author Style: 개발자가 지정한 스타일링

User Style: 사용자가 지정한 스타일링

Browser: 기본적으로 브라우저에서 정의된 스타일

Author Style -> User Style -> Browser 순으로,

정의해놓은 스타일이 없다면 Casacade(폭포, 위에서 아래쪽으로)식으로 스타일링 된다.

**Cascading의 연결고리를 끊어내는 딱 한가지!**

**!important**

css스타일을 정의할 때 !important를 쓰게 되면 가장 먼저 스타일링된다.

가급적 사용하지 않는 것이 좋다.

**2. 선택자(Selectors)**

Universal \*

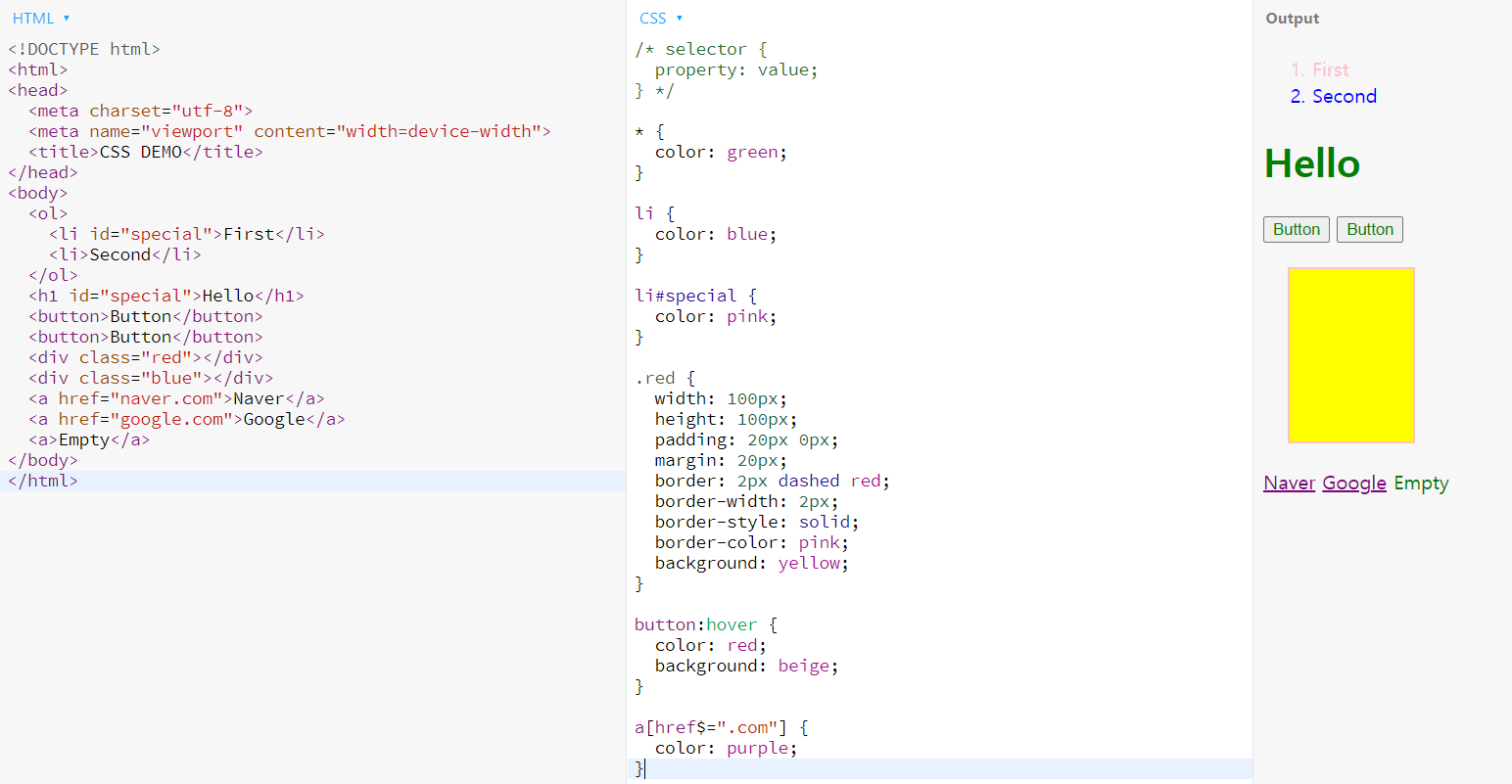
type Tag

ID #id

Class .class

State :

Attribute [ ]

#special // id이외에도 태그를 지정할수 있다. 위에서는 li만 css를 적용하였다.

padding 혹은 border를 적용할 때 attribute를 잘 알고있는 것이 좋다.

button:hover // hover는 커서를 버튼쪽으로 움직였을때를 지정하는 attribute이다.

[attribute]에서 a[href^=”naver”]이면 a태그의 href의 값이 naver로 시작하는 항목만 css 적용한다.

a[href$=”.com”]이면 a태그의 href의 값이 .com으로 끝나는 항목만 css 적용한다.

a[href\*=”asd”]이면 a태그의 href값에 asd가 포함되는 모든 항목을 css 적용한다.

**중요! -** 웹사이트를 만들 때 가장 중요한 것은?

**3. 우리가 만든 Box들을 원하는 위치에 원하는 사이즈로 배치하는 것이 중요하다!**

**Display**

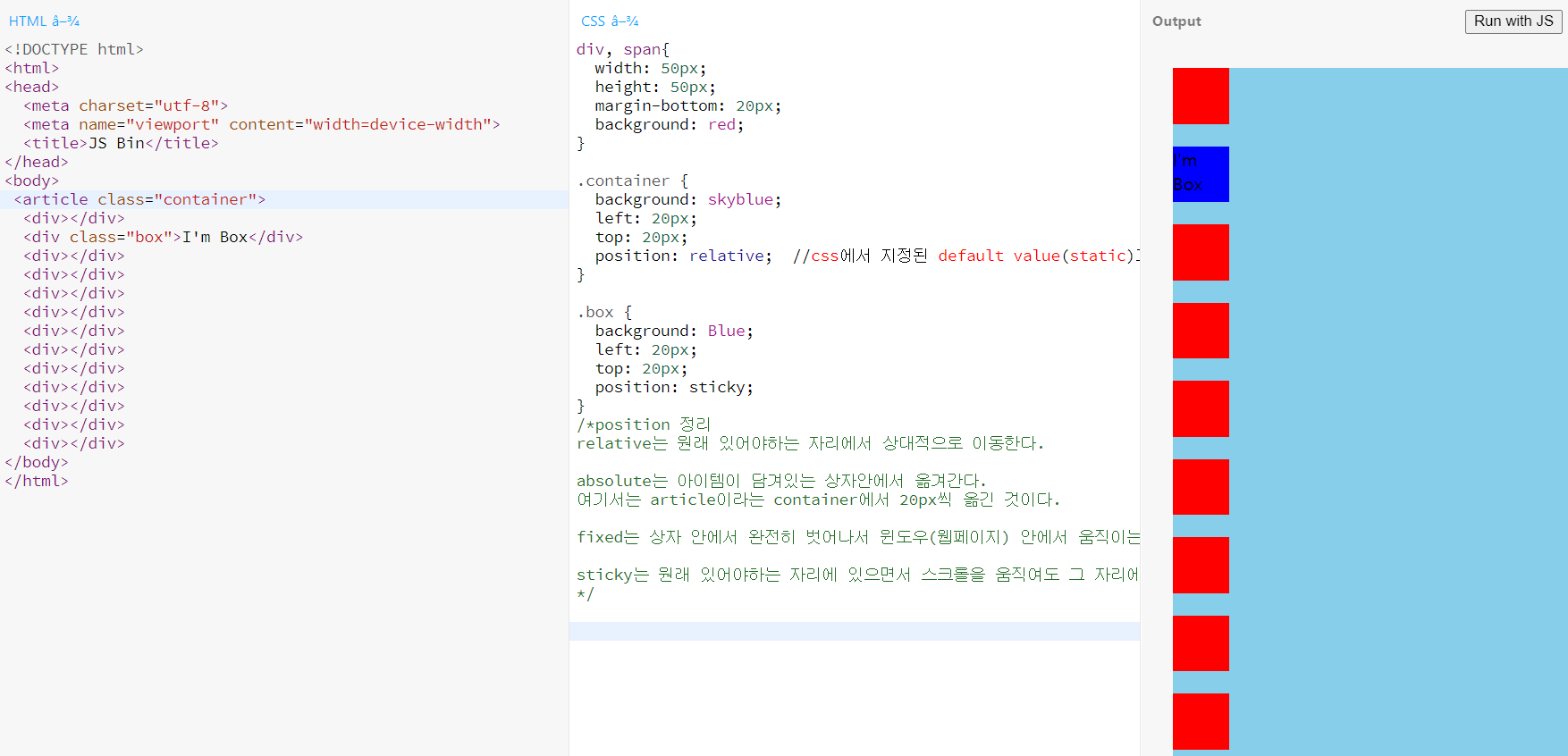


Diplay는 두가지 레벨로 나뉜다.

공간이 있으면 한줄에 출력하는 inline-block

각각 한줄을 차지하는 block

**Position**



**positon 정리**

-relative는 원래 있어야하는 자리에서 상대적으로 이동한다.

-absolute는 아이템이 담겨있는 상자안에서 위치가 변한다.

\*예)여기서는 article이라는 container에서 20px씩 옮긴 것이다.

**엘리쌤 코멘트(absolute)**

absolute를 쓰면 파란색 박스가 원래 있던 자리에서 쏘~옥 빠져나오기 때문에 빨간색 박스가 파란색이 있던 자리로 올라오는게 보이시죠? 그리고 absolute의 경우 부모 박스중에 가장 근접한 static이 아닌( relative, absolute, fixed) 부모 박스 포지션 기준으로 움직여요.

* absolute를 사용했을 때 css에서 부모 박스를 기준으로 움직인다.(원래 이해대로라면 absolute된 박스를 기준으로 20px씩 움직여야한다.)

-fixed는 상자 안에서 완전히 벗어나서 윈도우(웹페이지) 안에서 움직이는 것이다.

-sticky는 원래 있어야하는 자리에 있으면서 스크롤을 움직여도 그 자리에 그대로 있다.

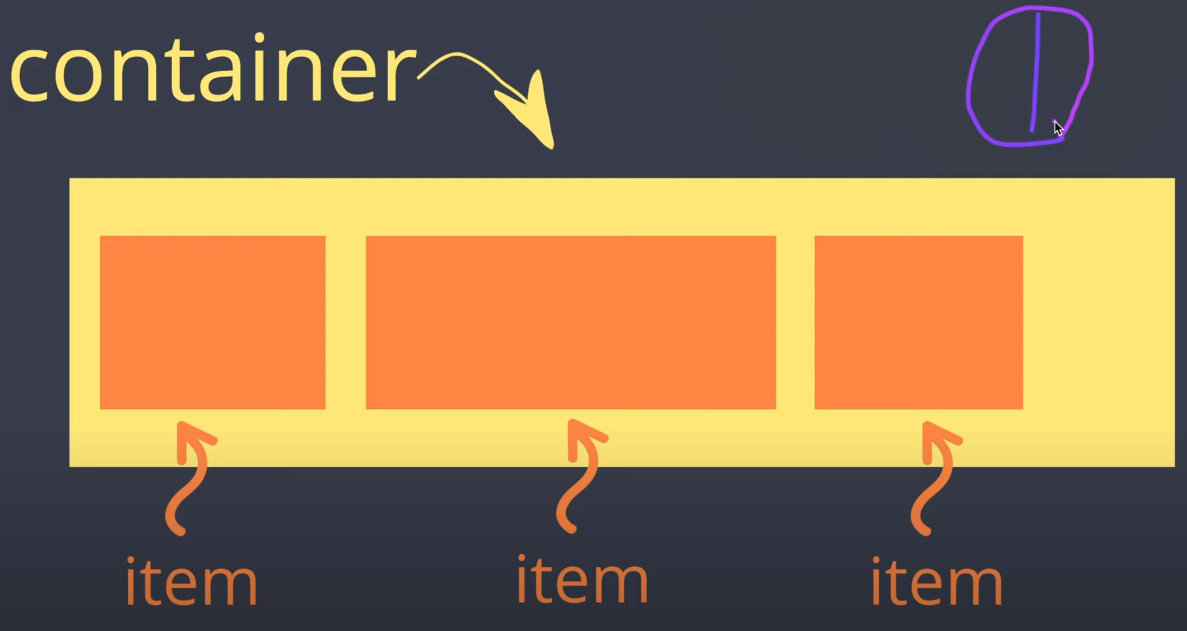
**새로운 기능을 쓸 때에는 모든 브라우져에서 사용할 수 있는 지 확인해야한다!!**

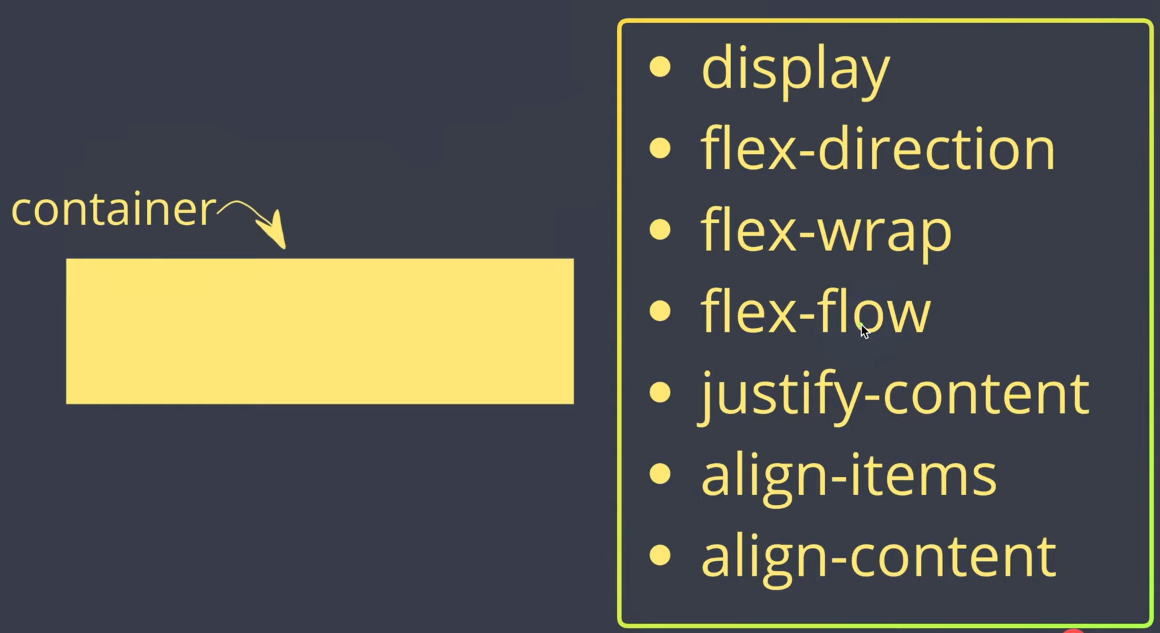
* <https://caniuse.com/>

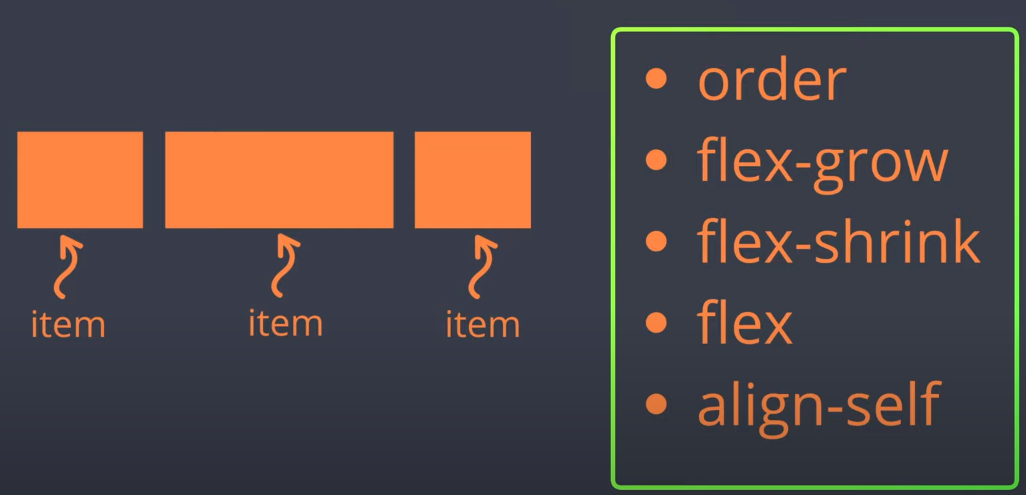
**4. CSS의 꽃: Flex Box  
Flexbox는 박스와 아이템들을 행과 열로 자유자재로 배치시켜줄 수 있는 유연한 아이!**

**1. Container**

**컨테이너(박스)에 적용되는 속성값이 존재한다. 또한 각각의 아이템들에 적용할 수 있는 속성값도 존재한다.**



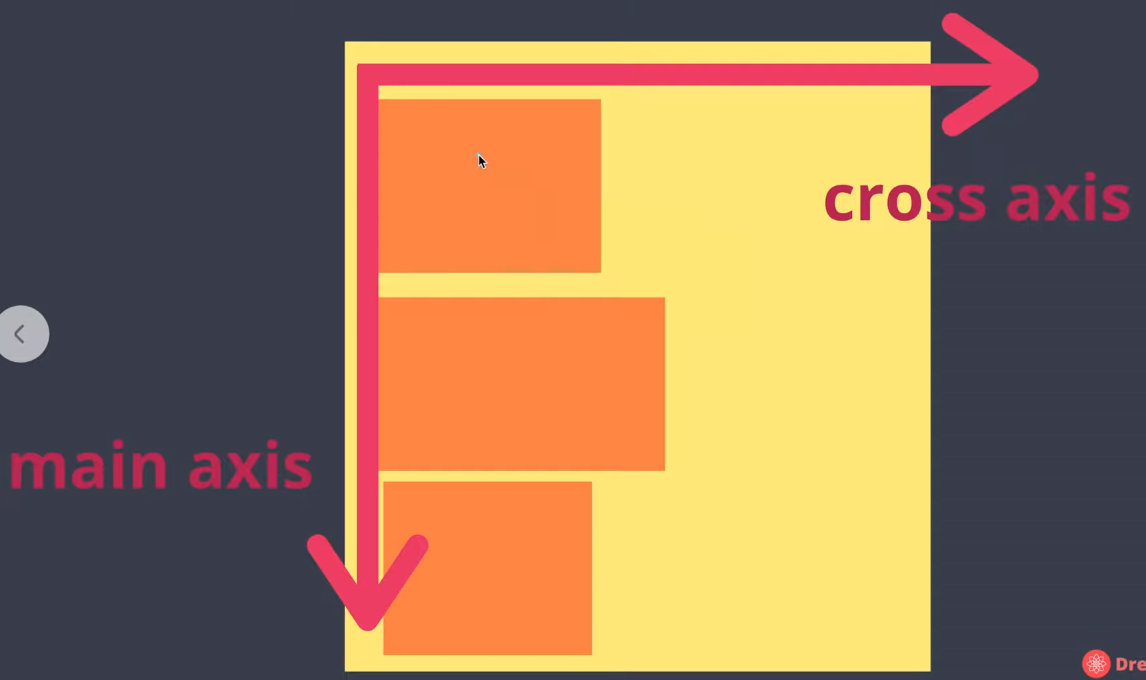




* Flexbox는 박스에 지정하는 속성값과 아이템에 지정하는 속성값이 각각 존재한다.

**2. Flexbox에는 중심축과 반대축이 있다.**



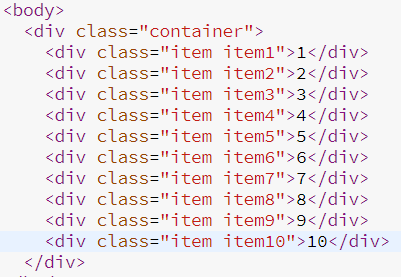


* **중심축을 수평, 수직을 선택하는 것에 따라 반대축이 바뀌게 된다.**

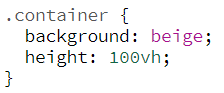
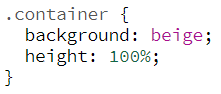
**\*꿀팁\***

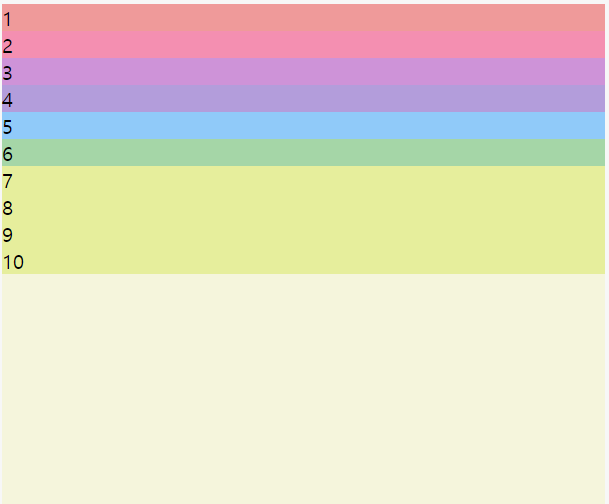
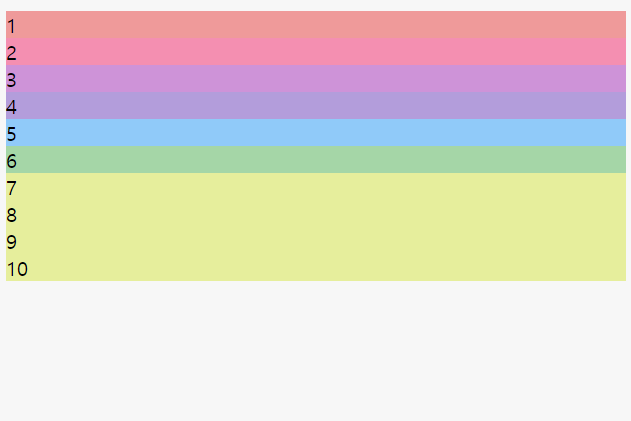
div.container>div.item.item${$}\*10

 + tab 키를 누르면 아래처럼 출력된다.



**100vh vs 100%**

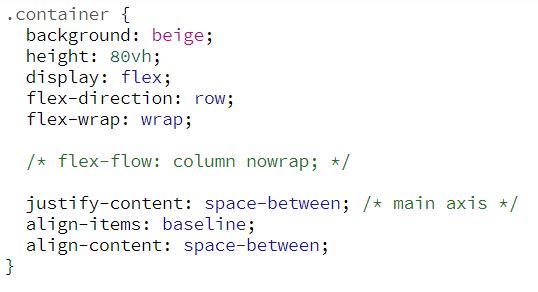
100vh는 부모 컨테이너(박스)에 상관없이 height를 지정한다.

100%는 박스에서 퍼센테이지로 height를 지정한다.

색깔 선정하기 좋은 사이트

* <https://material.io/resources/color/#!/?view.left=0&view.right=0>

**Container**



display: flex; // 컨테이너를 flexbox로 지정한다.

**flex-direction**: 컨테이너 안에서 아이템들의 정렬과 배치 방향 설정



**flex-wrap**: 아이템들 너비의 합이 컨테이너의 너비를 초과할 때 어떻게 처리되는 지를 설정



**flex-flow**는 flex-direction과 flex-wrap을 같이 설정할 수 있는 메소드

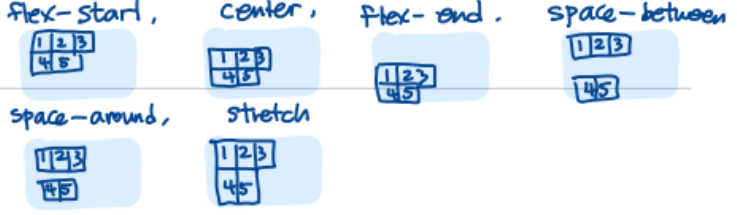
**justify-content**: 중심축에서 아이템을 어떻게 배치할 것인지 설정



**align-items**: 반대축에서 아이템을 어떻게 배치할 것인지 설정

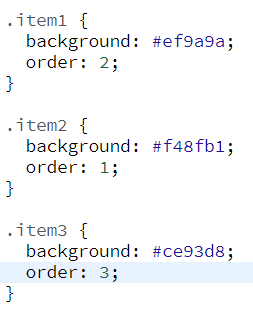
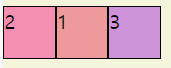


**align-content**: 아이템들을 한줄에 다 표시할 수 없기 때문에 다음 줄로 넘겼을 때 줄 사이의 여백을 설정한다.

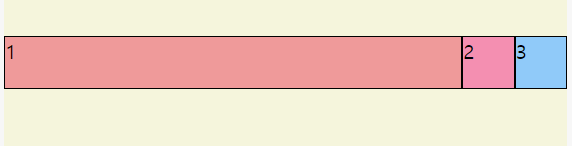
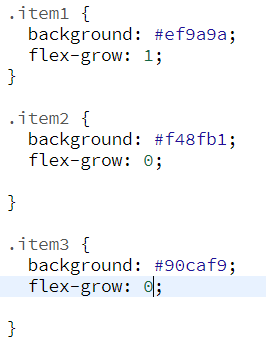


**item**

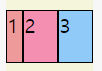
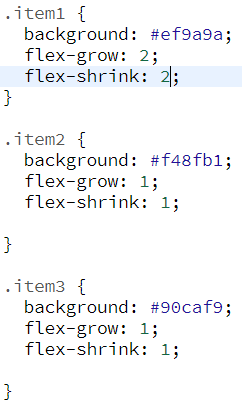
**order**: 각각의 아이템들의 위치를 설정한다. (많이 사용되지 않는다.)

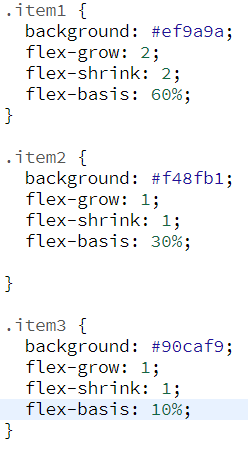
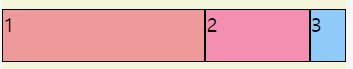
**flex-grow**: 웹페이지를 늘렸을 때 아이템의 늘어나는 사이즈의 비율 설정



**flex-shrink**: 웹페이지의 크기를 줄였을 때 아이템의 줄어드는 사이즈의 비율 설정



**flex-basis**: 아이템들의 크기를 비율로 설정(%, 퍼센테이지로 설정한다.)

**align-self**: 아이템별로 위치를 지정한다.

